Concept Design

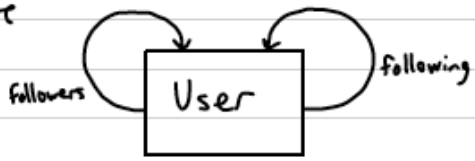
Concept

Following

Purpose

Create a connection from one user to another to express interest from the first user in the second, and distinguish the second user in the first user's feed

State



Action

- follow(u1: User, u2: User) : u1.following.add(u2.username);
  u2.followers.add(u1.username);
- unfollow(u1: User, u2: User) : u1.following.remove(u2.username);
  u2.followers.remove(u1.username);

OP

- all disj u1, u2: User | after follow(u1,u2) and not unfollow(u1,u2), u2.username in u1.following and u1.username in u2.followers and all f: freets authored by u2 | f highlighted in u1's feed

- all disj u1, u2: User | after follow(u1,u2) and unfollow(u1,u2), u2.username not in u1.following and u1.username not in u2.followers and all f: freets authored by u2 | f not highlighted in u1's feed

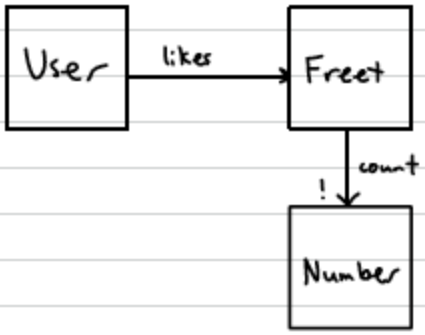
Concept

Upvote

Purpose

Track relative popularity of Freets, and sort feed based on decreasing relative popularity

State



Actions

- like (u : User, f : Freet) :
  u.likes.add(f.id);
  f.count ++;
- unlike(u : User, f : Freet) :
  u.likes.remove(f.id);
  f.count --;
- getCount(f : Freet) :
  return f.count;

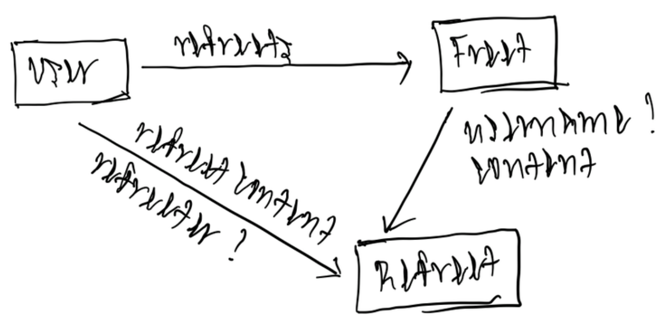
OP

- all u:User | all f: Freet | after k = count(f) and after like(u,f) and after k' = count(f), k' = k+1 and f.id in u.likes
- all u:User | all f: Freet | after k = count(f) and after like(u,f) and after unlike(u,f) and after k' = count(f), k' = k and f.id not in u.likes
- all u : User | all disj f1, f2 : Freet | if getCount(f2) > getCount(f1), f2 will appear before (higher) f1 in u's feed

Concept:
Refreeting

Purpose:
Allows users to share post of other users and comment their opinions of those freets

State:



Actions:

createRefreet(u: User, f: Freet, c: refreetContent):
    newFreet: createFreet(f.username, f.content, u, c);
    freetList.add(newFreet);

updateRefreet(c: refreetContent, u: User)
    toUpdate = freet in freetList if
        freet.id is correct and
        u === freet.refreeter;

    toUpdate.refreetContent = c;

deleteRefreet(u: User)
    freetList.delete(freet) if freet.id is correct

Operational Principle
all u: User | all f: Freet | c1, c2: refreetContent |
    after u.createRefreet(f, c1), u.updateRefreet(c2)
all u: User | all f: Freet | c: refreetContent |
    after u.createRefreet(f, c), u.deleteRefreet()

Design Decisions

1. If a user refreets a post and the post's original author then updates or deletes that post, the refreet does not change.

When thinking about refreeting, which involves a user reposting another user's freet, we wanted to ensure that user's opinions and messages were truly reflecting how the users wanted to engage with the feed. Therefore, if a user finds a freet interesting, they can refreet that post to share their commentary on that post, and that would be maintained. However, if an author of a freet also wants to delete or change their own freet, they have the capability to do so. To leverage these two interests, we decided to give authors and refreeters the ability to complete these actions.

Alternatives Considered: We considered deleting a refreet if the original author deleted their freet. However, we noticed how this would limit the scope of what users can post: they would have to only post original thought instead of sharing their opinions/commentary on someone else's thoughts. This would also prevent users from stealing other user's ideas that they saw in a post on the feed; they could just keep the content of that freet in their refreet and it is ensured that it stays that no matter what the original author does.

2. If a user tries to refreet a refreet, the new refreet will only show the original author's content and not also include the previous refreeter's content.

If multiple people thought that a freet was interesting, they could possibly cause a chain of refreets. However, with more and more users refreeting the same original freet, there could be less visibility of the original author's viewpoint and more so visibility of the opinions of the refreeters. Because the original author created a freet to share their original thought, they should be more centered when a freet is being reposted so we decided to have only the original freet's content and the new user's comment in a refreet.

Alternatives Considered: We considered allowing a large chain of refreets to occur where there could be some dialogue garnered by the comments of multiple users on

an original freet. However, it would potentially cause user's viewing the feed to focus more on the chain of conversations instead of the message that the original user wanted to get across, since it could be more obstructed by people's various levels of support and opinions on the original freet.

3. The feed is sorted based on posts with the most likes.

We decided to sort the feed based on the post with the most likes such that we can display the most popular freets first. By having the feed sorted by popularity of the post, we can have more user engagement (by attracting potential new users to the app). It would also shed light on the types of post that users tend to like on the app which would allow the app to display the culture of the community that the app mostly caters to.

Alternatives Considered: We considered having the feed sorted by when a freet was posted. Although this would be consistent with reflecting what users wanted to post at certain points in time, certain posts might not spark interest in users compared to others. We decided to place popularity before timeliness in order to attract the most users even though it may mean that the feed does not show what users are thinking currently.

4. A freet is highlighted if the author or refreeter of that freet is in your following list.

We decided to highlight freets when the user that is currently in session has the author or refreeter of that freet in their following list. We used highlighting as a way to distinguish the content has been posted that the current user is interested in from the rest of the posts in the feed. Therefore, the current user in session would have an easier time looking at the content that they believe would be most interesting to them. Since they decided to follow these individual users, they have an interest in their content.

Alternatives Considered: We considered moving the post that were authored or refreeted by users in the user in session's following list but this would conflict with our previous decision to prioritize the posts with the most likes. In order to give the user the ability to see the most popular post while also giving them the ability to seek the

posts by the users that they are most interested in, we sort the feed by the posts with the most likes and highlight the content written by users who are in the following list.

Social / Ethical Implications

1. Keeping refreets static

If user1 refreets user2's freet, any edits or deletions that user2 makes to that original freet are not transferred to user1's refreet. This has two clear implications. The first is that user1 does not need to worry about user2 changing the content of the freet to something that they do not resonate with. For example, if user2 changed their freet content from something harmless to something hateful, user1 will not be tethered to this new hateful message, as they refreeted the original harmless message. This is beneficial, as user1 can refreet a freet without needing to worry about the downstream effects of the freet being edited to something they do not agree with. However, the other implication is that user2 has a permanent, immutable record of their thoughts on the internet. That is, if they have changed their mind about something they freeted in the past but user1 already refreeted that freet, even if user2 changes the freet, the original copy still lives on user1's page, and cannot be taken down unless user1 decides to delete the refreet. This can be problematic—if user2 posted something hateful or offensive in the past but has had a change of heart and tries to edit or delete the freet, an original copy of their previously offensive words live on the internet if another user refreeted this original freet. One way to mitigate this would be to update the refreet to something like "The original freet has been modified/deleted." This way, user1 does not need to worry about edits to a freet showing up in their refreets, but user2 does not need to worry about a permanent copy of their words living on the internet.

2. Your followers/following are only visible to you.

Your followers/following lists are only available on your profile page, which are only accessible if you are signed in as that user. Thus, you cannot see the followers/following lists of any other users. In some ways, this has positive implications: this enhances privacy as many users do not want their lists of followers/following available to the public. However, this also has a negative

implication: it is difficult to create community on the Fritter platform. Often times, users find each other via mutual followers, or users check their friends' following/followers lists to find people they may know. This feature is not possible with our current implementation, as your followers/following lists are only available to you. One way to mitigate this would be to have an opt-in system for displaying your followers/following lists. Many social media apps have this to some extent: for example Instagram allows you to set your account to private (does not display followers/following to anyone except your followers) or public (anyone can see your followers/following regardless of whether they follow you). This would allow users to have autonomy in their privacy: if they're okay with displaying their followers, then they can opt in, otherwise they do not need to opt in.

3. Feed sorted in order of descending likes

Freets are sorted by descending number of likes. That is, freets with the most likes appear at the top, while freets with the least likes appear at the bottom. The idea behind this was so that content that is more relevant/interacted with appears first, as it likely would interest the user scrolling through their feed more. However, this also makes it difficult for freets with little interaction, or freets that were recently posted, to gain popularity. If a freet has little interaction, or a freet was just posted (initialized to 0 likes), it appears at the bottom of the feed. Users are less likely to scroll to the bottom of the feed, meaning these freets that are already not popular receive less interaction, and this creates a feedback loop. This disadvantages users with a small following who may be trying to raise awareness on a certain event or cause. There are several ways to mitigate this. One way would be to use a machine learning powered recommender system. That is, freets that pertain to a particular user's interests may appear at the top of that specific user's feed, regardless of how many likes it has. This would require a lot of extra data about the user, however, and may be both financially and computationally expensive to implement. Another option would be to perhaps alternate between highly popular freets and not popular freets in the feed, however this may make the user experience less smooth and more choppy, which would not be ideal.